INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Image Retrieval by Range Query Composition of Region Categories

Julien Fauqueur — Nozha Boujemaa

N° 4686

Decembre 2002

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4686--FR+ENG

Image Retrieval by Range Query Composition of Region Categories

Julien Fauqueur , Nozha Boujemaa

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Imedia

Rapport de recherche n° 4686 — Decembre 2002 — 30 pages

Abstract: In this paper, we present an original and new framework to allow complex multiple regions queries based on composition of regions categories. This problematic has merely not been addressed in the litterature even though it corresponds to photo-agency usage scenario.

We first present the image segmentation technique specifically designed for image retrieval by regions based on the clustering of Local Distributions of Quantized Colors (LDQC).

Then, categories of similar regions are formed automatically by clustering their visual features. The representative regions for each category give to the user an overview of all the regions. They allow him to query the database by specifying different types of regions which should appear in the retrieved images but also types which shouldn't. This enables our system to retrieve images from complex queries of region categories composition. This query mode differs completely from the classic Query-by-Example paradigm since there is no query image. Only the user has a clear idea of what he is looking for.

Our retrieval system handles region range query by considering regions from the same category but also from neighbor categories.

The resulting indexing and retrieval implementation turns out to be simple and very fast even on very complex query compositions and large image database. It was tested on a database of 9,995 images from the Corel image database.

Key-words: image composition, region categories, image segmentation, clustering, multiple region query, range query, user interface, visual semantics

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Recherche d'images par composition de catégories de régions

Résumé : Dans ce rapport, nous présentons un cadre de recherche d'images original et nouveau permettant des requêtes complexes par régions multiples basée sur la composition de catégories de régions. Cette problématique n'a pas encore été abordée dans la littérature bien que le scénario d'usage corresponde à un besoin effectif de type agence-photo.

Nous présentons tout d'abord la technique de segmentation d'images conçue spécifiquement pour la recherche d'images par régions, basée sur la classification des distributions locales de couleurs quantifiées (ou LDQC, pour Local Distributions of Quantized Colors).

Puis, nous détaillerons la formation automatique des catégories de régions par regroupement de leurs caractéristiques visuelles. Les régions représentatives des différentes catégories fournissent à l'utilisateur un résumé des régions de la base. Elles lui permettent de formuler sa requête en spécifiant les différents types de régions devant apparaître dans les images retrouvées mais aussi les types de régions indésirables. Le système proposé peut alors retrouver les images à partir de requêtes complexes sur la composition des catégories de régions. Ce mode de requête contraste avec le paradigme classique de recherche par l'exemple dans le sens où il n'existe pas d'image requête. Seul l'utilisateur a une idée claire de ce qu'il recherche. Le système supporte par ailleurs les requêtes par intervalles en considérant les régions de la même catégorie mais aussi celles des catégories voisines.

L'approche d'indexation et de recherche s'avère simple et très rapide, même pour des requêtes très complexes sur de grandes bases d'images. Les tests ont été effectués sur une base de 9995 images de la base d'images Corel.

Mots-clés : composition d'image, catégories de régions, segmentation d'image, classification, requête régions multiples, "range query", interface utilisateur, sémantique visuelle

1 Introduction

The earliest Content-Based Image Retrieval approach is the global query-by-example approach, initially developped in these systems: QBic [1], PhotoBook [2] or Virage [3] and others. For a more detailed study of a state-of-the-art we can refer to [4].

But most often the user wants to retrieve similar objects (or image parts) rather than images viewed as a whole. In this context of query the global approach is restrictive and unsatisfactory. In database of natural composite images, the search for similar objects using image global features is highly biased by surrounding objects and background. To minimize this difference between user intention and the retrieved images (usually called *semantic gap*) the relevance feedback mechanism was strongly investigated in the litterature to allow user express his preference to the system. In our point of view, relevance feedback is not the only way to take into account user preference but partial query formulation also allow user to specify which part of the image is the target of his interest.

Designing a region-based query system remains a challenging and open problem: automatic detection of regions of interest is a hard task and the retrieval process must deal with the increased number of entries in the database.

Among these few systems we can cite the Blobworld [5] and Netra [6]. In Blobworld, region segmentation is performed by clustering joint color and texture vectors with the expectation/maximization (EM) technique. Segmentation is approximate and many small areas are omitted. In Netra, segmentation is contour-based and provides satisfactory regions but is very time consuming.
Both systems simply perform an exhaustive search among regions in database from a single example region.

Image retrieval by multiple region query allows more specific searches than by single region query and global image query. This approach has been merely not addressed so far despite its obvious relevance to integrate more visual semantics in the query by image content.
A more recent technique involving region-matching for image retrieval is proposed in the SIMPLIcity system [7]. The similarity between two images is measured as combination of similarities between the regions which constitute both images. But the system actually performs *global image retrieval* since features of all regions in images are involved. The quality of region segmentation is not their main concern. Exhaustive image search is performed among one of 3 image categories: graph, textured and non-textured.
In VisualSeek [8], a multiple region query was proposed by sketching rectangles of synthetic colors. Image matching is performed by color set back projection at retrieval time which is computationally expensive.

Two major problems are recurring in existing region-based image retrieval systems: slow exhaustive search among database regions and tedious selection of example regions.

So far, all the focus was dedicated to the precision performance but at computation cost expense. The computational complexity of an image retrieval by region system is increased because of the existence of many regions per image which increases the number of entries in the database and the number of matches. The complexity grows even more in the case of multiple region queries. So there is a strong need for an index structuring scheme dedicated to multiple region query in large image databases.

We'll see that the approach of our new framework is very different and the two above mentioned problems are naturally solved.

In this paper we will present the three major aspects of our framework for image retrieval by region categories composition.

We will first detail in section 2 the Competitive Agglomeration algorithm which is used in different parts of our system to cluster visual features.

We'll propose in section 3 the **region extraction** scheme designed to meet the requirements of a region query system. It relies on a fast segmentation technique which detects coarse and relevant regions for the user. The image segmentation scheme proposed is based on the clustering of local color distributions evaluated over large neighborhoods.

Regions integrate more intrinsic variability to be visually more characteristic in the database.

In section 4, the generation of **categories of similar regions** by grouping their visual features will be explained. To allow range queries, we will also define their **neighbor categories**. From this point on, regions won't be considered individually anymore but will be totally identified to the category they belong to.

Then, in section 5, we will detail the approach to achieve **image retrieval by composition of region categories**. We will propose a formalism of query composition for our framework. Range query on region categories will also be introduced. We will present an adapted retrieval strategy to handle queries from complex compositions and then an efficient indexing and retrieval scheme.

In section 6, we will present an original **user interface** along with the **results**. the query interface will provide an overview of regions categories of the database and thus avoid tedious random browsing. The user will select among them those which will be the "positive query" or "negative query" examples. Complex query compositions will be easy to formulate and iterative query refinement will allow narrowing the search.

In section 7 **concluding remarks** on the specificity of this new multiple region query framework[1] and future work will be addressed.

[1] a detailed version of this work is published in an INRIA research report [9]

2 Visual Feature grouping method

For both region extraction and region categories generation, an efficient clustering scheme is required. The Competitive Agglomeration clustering, originally presented in [10], was chosen because of its major advantage to determine automatically the number of clusters. It is an advanced extension of FCM (Fuzzy C-Means) [11] clustering algorithm.

Using notations from [10], we call $\{x_j, \forall j \in \{1, ..., N\}\}$ the set of N data we want to clusterize and C the number of clusters. $\{\beta_i, \forall i \in \{1, ..., C\}\}$ denote the prototypes to be determined. The distance between data x_j and prototype β_i is $d(x_j, \beta_i)$. The CA-clustering is performed by minimizing the following quantity J:

$$J = J_1 + \alpha J_2, \tag{1}$$

where

$$J_1 = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i) \tag{2}$$

and

$$J_2 = -\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2 \tag{3}$$

Subject to membership constraint:

$$\sum_{i=1}^{C} u_{ij} = 1, \forall j \in \{1, ..., N\} \tag{4}$$

where u_{ij} represents the membership degree of feature x_j to prototype β_i. Minimizing J_1 separately is equivalent to perform an FCM clustering which determines C optimal prototypes and the fuzzy partition U given x_j and C using distance d. Therefore J is written as a combination of two opposite effect terms J_1 and J_2. So minimizing J with an over-specified number of initial clusters classifies data and simultaneously optimizes the number of classes.

J is minimized recursively. α is the competition weight and should allow a balance between terms J_1 and J_2 in equation (1). At iteration k, weight α is expressed as :

$$\alpha(k) = \eta_0 \exp(\frac{-k}{\tau}) \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i)}{\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2} \tag{5}$$

As iterations go, α decreases so emphasis is first given to agglomeration process, then to clustering optimization. α is fully determined by parameters η_0 and τ.

During the algorithm spurious clusters are discarded. Spurious clusters are those whose population, defined by quantity $\sum_{j=1}^{N} u_{ij}$ for a cluster i, is below a given threshold. Convergence is decided when prototypes are stable. The clustering granularity is controlled by factor α through its magnitude η_0 and its decline strength with τ. The higher η_0 and τ, the higher α, so the more classes are merged. So for a given clustering granularity, CA determines the optimal number of classes.

CA will be used at three steps in our framework with different levels of granularity and different input feature vectors: for segmentation, it'll be used for image quantization then to perform the actual segmentation by local color distributions clustering and eventually to form the groups of similar regions using they feature vectors.

3 Region extraction

Detected regions should encompass a certain visual diversity to be visually characteristic, using a coarse segmentation. We want to stay beyond a too fine level of spatial and feature details. Besides the segmentation technique should be fast and unsupervised. Time is an important factor since we want to deal with large image databases. But fast and coarse shouldn't mean poor in quality: resulting regions should be intuitive for the user.

The Blobworld segmentation [12] is based on the clustering with EM algorithm of feature vectors extracted for each pixel. They consist of 8 components: 3 for color (Lab color space), 3 for texture (anisotropy, polarity, contrast) and 2 for position (x and y). EM has the drawback to assume the number of clusters known in advance. Detected regions are chaotic and many small areas appear to be discarded inside bigger coherent regions. Netra system uses the Edgeflow contour-based approach [13]. Extracted regions are satisfactory but the technique takes about 50 seconds per image (about twenty times longer than our technique on the same machine and same images). EdgeFlow program was tested with the available binary program on the web[2].

Our segmentation approach is based on the CA-clustering of local color distributions of the image. This feature naturally integrates the diversity of colors in pixels neighborhood. Besides the clustering of a feature vector representative of a large neighborhood provides coherent regions more naturally: indeed forming regions by grouping large similar neighborhoods leads to coarser regions than grouping pixels or very small neighborhoods, since less spatial postprocessing is required.

The choice of the color set to compute local distributions is crucial: it must be compact to gain speed in clustering and representative of a pixel neighborhood. We define the color set as the adaptive set obtained by image color quantization, to dramatically reduce the number of colors to process without losing too much perceptual information. Then for all neighborhoods in the image, local distributions are evaluated on this small and relevant color set. We'll refer to them as Local Distributions of Quantized Colors (*LDQC's*). After clustering, LDQC prototypes are back projected onto image, then small regions are either merged or discarded.

The three segmentation steps are the following:

[2] http://vision.ece.ucsb.edu/segmentation/edgeflow/software/index.html

- color quantization (provides the *quantized colors*)
- clustering of local distributions of quantized colors (provides the *prototypes of LDQC's*)
- merge and removal of small regions

3.1 Color quantization

Image is quantized in the Luv color space by CA-clustering of color pixels using the Euclidean distance L^2. This distance in the Luv color space is a good approximation of the human eye perceptual similarity between colors. The clustering granularity of CA was chosen such that big areas in images with a strong texture are represented by more than one color. At clustering convergence the color prototypes define a set of n quantized colors. Since CA determines automatically the number of clusters, the number of quantized colors n will be representative of the color diversity of the natural images.

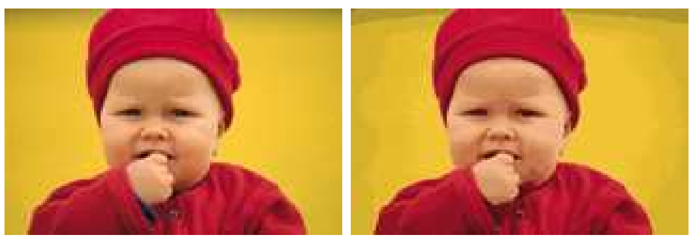

Figure 1: original image (left) and quantized image (right)

3.2 Grouping of LDQC adaptative image feature

To determine all the LDQC's in the image, we slide a window over pixels and evaluate the corresponding local distribution of quantized colors. Let's denote S_W the window surface and S_{TOT} the image surface. LDQC's are evaluated every wr pixels, where wr is the window radius.

An appropriate distribution measure should be used for the clustering. L^p distances are widely used for global color distributions on uniformly subsamples color spaces but are inaccurate for small adaptive color sets as those obtained on neighborhoods.

The color quadratic form distance [14] provides a precise distance to compare any kind of color distributions by integrating the inter-bin color similarity. Let's consider X and Y two local distributions over the n quantized colors and write them as pairs of color/percentage:

$X = \{(c_1, p_1^X), ..., (c_n, p_n^X)\}$ and $Y = \{(c_1, p_1^Y), ..., (c_n, p_n^Y)\}$

Then the quadratic distance is expressed as:

$$\begin{aligned} d_q(X,Y)^2 &= (X-Y)^T A (X-Y) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(p_i^X - p_i^Y)(p_j^X - p_j^Y)a_{ij} \end{aligned} \tag{6}$$

where $A = [a_{ij}]$ is the matrix of similarities a_{ij} between colors c_i and c_j:

$$a_{ij} = (1 - d_{ij}/d_{max}) \tag{7}$$

where d_{ij} is the Euclidean distance in Luv space and d_{max} the maximum of this distance in the color space.

After clustering, the segmented image is obtained by assigning to the S_{TOT}/wr pixels the label of the LDQC prototype which minimizes the quadratic distance to the LDQC centered on that pixel. A maximum vote filter is applied to discard isolated pixels.

Window surface S_W defines the spatial resolution of the segmentation: the higher S_W and the bigger patterns we extract. wr was set to 16 (on average) pixels for a 384x256 image.

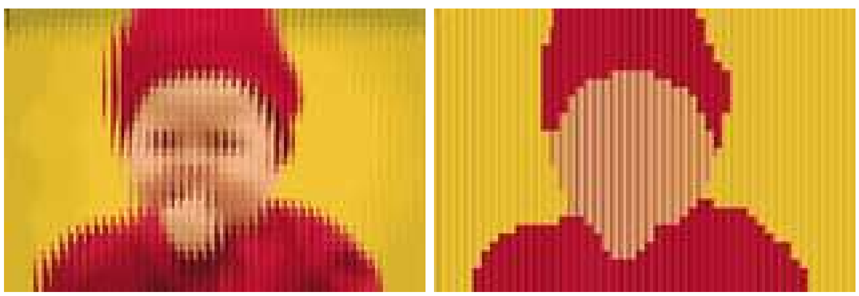

Figure 2: image of LDQC's (left) and image of LDQC prototypes after clustering (right). These two images are created to visualization all the LDQC's in the image: each square neighborhood is replaced by the corresponding local color distribution drawn with color bars of width proportionnal to the quantized color population.

3.3 Adjacency information

At this point, we obtain a complete partition of the image into adjacent regions. Some regions may be too small to constitute regions of interest and they increase needlessly the total number of regions in the database. Besides, in complex scenes, they're often located at the frontier between two big regions or inside a big region. They should be merged to improve the topology of regions of interest.

We want final regions of interest to cover a minimum of 1.5% of the image surface. Below this surface a region is merged to its closest visual neighbor region if it has one and

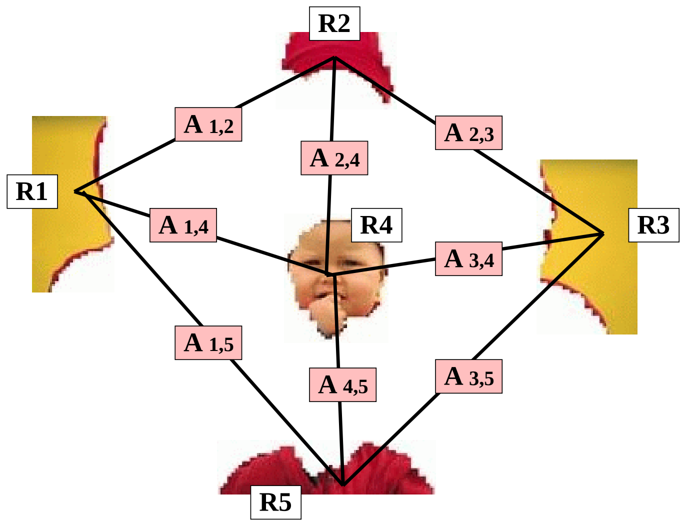

Figure 3: RAG structure of the partionned image

Figure 4: final segmented image represented with the original mean colors of each region

is discarded otherwise. Two regions are said to be visually close if they have close mean quantized color distributions. Remaining small regions are discarded and not indexed.

Region merge is made possible by the use of a Region Adjacency Graph structure [15] (RAG). Region attributes (surface, color distribution, contours, barycentre) are stored in the nodes and region adjacency information in the graph edges (adjacency, common contours length).

4 Categories of regions and neighbor categories

Now we're provided with the extracted regions from all images in the database. These regions can be seen as the salient constituting units of each image.

The categories of regions are defined as the clusters the regions which have similar visual features. This will be the basis of the definition of similar regions in the retrieval phase.

We choose to simply characterize regions by their mean color in the LUV space. This 3D-descriptor has the advantage to represent color which is the most commonly used cue to perform visual search and to allow visualization of regions in the feature space as illustrated in figure 5 below.

It has the advantage to be robust and generic. No restriction is made concerning the choice of the descriptor as long as it is provided with a metric required by the clustering phase.

To visualize the regions in the feature space (LUV color space), we represent the regions thumbnails in a VRML scene in which their 3D location corresponds to their mean color. A screenshot of this VRML scene is shown in figure 5. We must note that regions are more suited than images for such a representation than images since they are by construction visually more homogenous than entire images.

The overall appearance of the regions thumbnails laid in the 3D scene is rather continuous in terms of color changes. The global shape of the data set is quite compact and has peaks pointing to saturated colors corresponding to red, green, blue, and also to black and white.

Since data are compact and dense, no natural data grouping can be expected. We can not make a priori assumption concerning the well-definition of clusters of regions for any database. But clustering the features will provide better categories of regions than performing a systematic subdivision of the color space because clustering is data-driven.
What can be guaranteed though is a homogeneity within obtained categories by setting a fine clustering granularity.

Since similarity between regions will be defined, at a first level, as members of the same category, a fine clustering granularity ensures the retrieval of very close regions (hence high retrieval precision). But in our region range query scheme we will also want to take into account regions contained in close categories (called "neighbor categories") to also allow high recall. The **neighbor category** of a category C_q of prototype p_q is defined as a category C_j whose prototype p_j satisfies $d(p_q, p_j) \leq \gamma$, for a given range radius threshold γ.

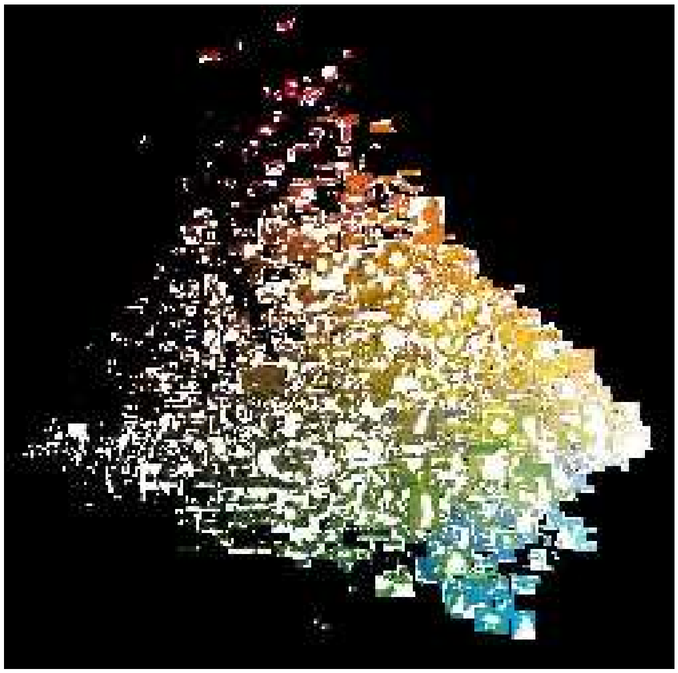

Figure 5: Regions thumbnails in the LUV color space. Their location corresponds to their mean color position. Thumbnails are cropped images of regions over white background. For visualization sake, only 5500 regions are shown.

We call $N^\gamma(C_q)$ the **set of neighbor categories** of a category C_q. By convention, we decide that category C_q itself always belongs to $N^\gamma(C_q)$ as a neighbor of itself at distance zero. γ will be used to adjust the range value at retrieval time.

The combination of homogeneous region categories with the integration of neighbors categories is a key choice in the definition of the range query scheme.

With a fine granularity, CA algorithm (as detailed in section 2) is used to clusterize all the database regions mean color features. It provides the region categories along with their prototypes. In this case, a prototype is a mean color which may be not necessarily be represented by a region in the database. So for each category, we rather define its **representative region** as the region in the category whose mean color is the closest to the category prototype. The category representatives will only be used in our system in the query interface to identify each category.

5 Image retrieval by composition of region categories

In this section we are going to detail the different parts to achieve image retrieval by region category compositions: expression and model of the different query compositions and the adapted indexing and retrieval scheme.

5.1 Query composition

We have now defined the categories of similar regions. For each category we also have defined its representative region and its neighbor categories which depend on the range radius.

The representative regions of all categories are meant to provide an overview of the available regions in the database, as we'll see in section 5.2. They will be used in the user interface to represent each category to query the database. With the help of the database representative regions, the user will select positive query categories (referred to as **PQC**s) and negative query categories (referred to as **NQC**s).

A **query composition** will then be defined as a query formulation such as: "find images containing regions in these PQCs and no region from those NQCs" in its most complex form.

This section will detail the formal approach used to perform such query composition by starting with a simple composition and growing its complexity. We will see at each step the expression of the set of images in the database which match the composition.

5.1.1 notations

Region categories: Let $C_1, C_2, ..., C_P$ be the region categories obtained in previous section with CA. We denote by $p_1, p_2, ..., p_P$ its feature prototypes.

Sets of images: Since the query will consist in retrieving image which contains certain types of regions (i.e. regions of a specific category C), we need to define $IC(C)$ to be the set of images containing at least one region belonging to category C.

Positive Query Category: It corresponds to a user-selected category of regions which should appear in retrieved images. It was introduced earlier as PQC. The PQCs will be represented as the list of their labels $\{pq_1, pq_2, ..., pq_M\}$ (e.g. C_{pq_1} can be a category approximately corresponding to skin regions, C_{pq_2} to sky regions, and so on...).

Negative Query Categories: It corresponds to a user-selected category of regions which should *not* appear in retrieved images. It was introduced earlier as NQC. The NQCs will be represented as the list of their labels $\{nq_1, nq_2, ..., nq_R\}$.

Sets of retrieved images: It contains images of the database which have a relevant composition of region categories and is denoted: S_{result}.

5.1.2 basic query composition: "find images with a region in this PQC"

The most basic query consists in retrieving images in the database which contain a region from a single PQC category denoted C_q say. Then the set of retrieved images is simply written as: $S_{results} = IC(C_q)$.

To carry a region range query, we want to take into account neighbor categories. When searching images with regions in the PQC C_q we'll search images with a region in C_q **or** with a region in the neighbor categories of C_q, for a given γ. The **or** conjunction is translated as a **union** of image sets:

$$S_Q = \bigcup_{C \in N^{\gamma}(C_q)} IC(C) \tag{8}$$

This constitutes the basis of the region range query.

The influence of range radius γ on the definition of neighbor categories is illustrated in figure 6.

In the rest of the paper, we'll perform retrieval from PQCs and NQCs and their respective neighbors.

5.1.3 positive query composition: "find images with regions in these PQCs"

Now we extend the query to more than one PQCs. Instead of dealing with a single PQC C_q we assume we have M PQCs: $C_{pq_1}, C_{pq_2}, ..., C_{pq_M}$. We search images which have a region in C_{pq_1} or its neighbors **and** a region in C_{pq_2} or its neighbors **and** .... **and** a region in C_{pq_M} or its neighbors. The **and** conjunction is translated as an **intersection** of image sets obtained in equation 8:

$$S_Q = \bigcap_{i=1}^{M} \left[\bigcup_{C \in N^{\gamma}(C_{pq_i})} IC(C) \right] \tag{9}$$

The set S_{result} constitutes the first level of relevant images:

$$S_{result} = S_Q \tag{10}$$

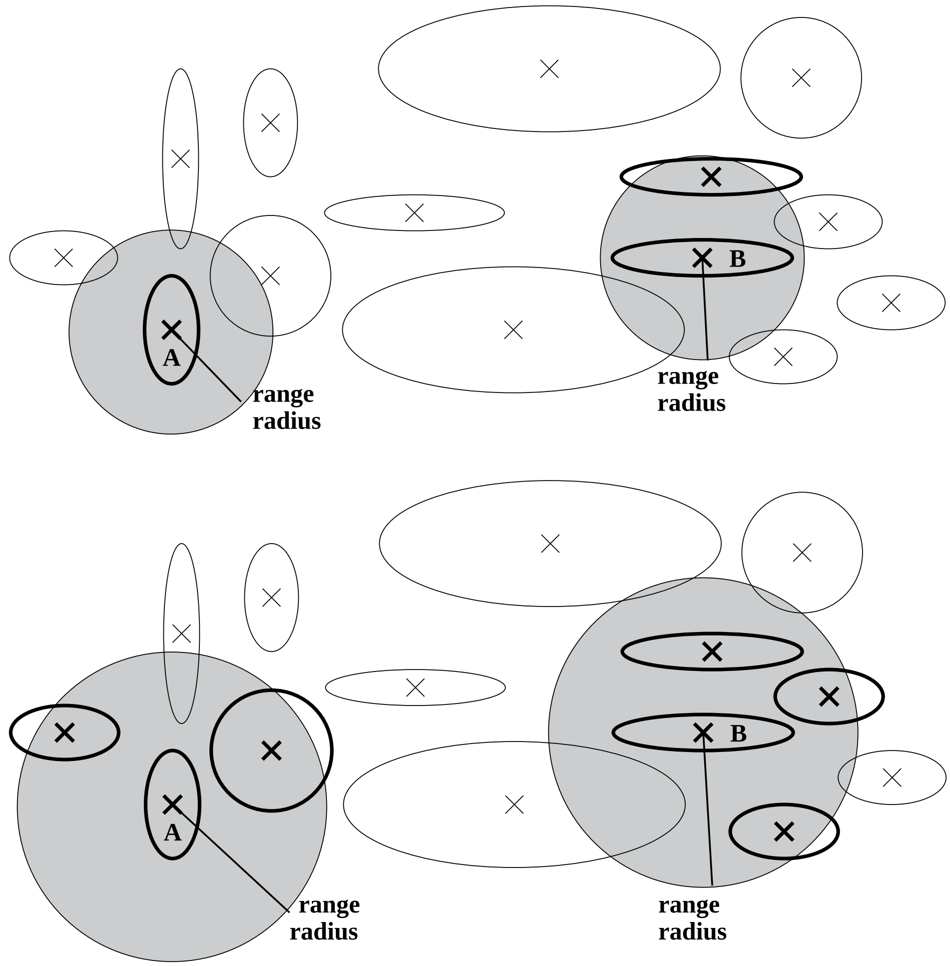

Figure 6: **Range radius and neighbor categories**. A and B are two PQCs. Influence of range radius γ on definition of neighbor categories: a high radius (top figure) and a lower radius (bottom figure) will integrate more or less neighbor categories to define the type regions searched. The grey disks of radius γ cover the neighbor categories. Neighbor categories are drawn with thicker contours than other categories. Prototypes are identified by crosses.

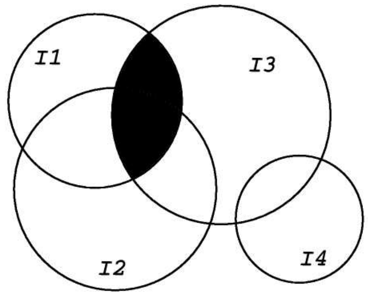

Figure 7: **Positive query composition as intersection of image sets:** Intersection of the sets of images I_1 and I_3 which respectively contain a region of category 1 and a region of category 3.

5.1.4 negative query composition: "find images with regions in these PQCs but no regions in those NQCs"

As before, their neighbor categories should also be taken into account to allow an adjustable similarity level in the search of NQCs. So the set S_{NQ} of images containing the NQCs is written as above:

$$S_{NQ} = \bigcap_{i=1}^{R} \Big[\bigcup_{C \in N^{\gamma}(C_{nq_i})} IC(C) \Big] \tag{11}$$

The query formulation now consists in a list of PQC labels $\{pq_1, pq_2, ..., pq_M\}$ and NQC labels $\{nq_1, nq_2, ..., nq_R\}$. So the set S_{result} of retrieved images which have regions in the different PQCs and which don't have regions in the NQCs is expressed as the set substraction of S_Q and S_{NQ}:

$$S_{result} = S_Q \setminus S_{NQ} \tag{12}$$

This set S_{result} constitutes the second level of relevant images.

5.1.5 exclusive query composition: "find images which *exclusively contain* regions in these PQCs"

This last query composition is the most restrictive since images which contain any region which does not belong to any of the PQCs nor any of its neighbors will be rejected. We remark that it is equivalent to the previous formulation in which NQCs would be all but the PQCs and their neighbors. This is interpreted as an exclusive search on the PQCs. We denote by S_{XQ} the set images which contain any region which does not belong to any of the

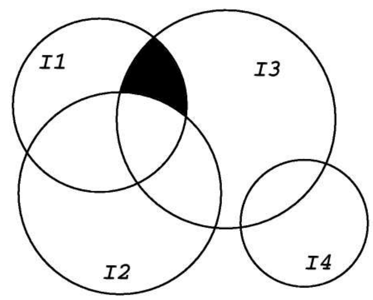

Figure 8: **Negative query composition:** Intersection of the sets of images I_1 and I_3 which respectively contain a region of category 1 and 3 and don't contain a region of category 2.

PQCs nor any of its neighbors:

$$S_{XQ} = \bigcap_{C \in \{C_1, C_2, \ldots, C_P\} \setminus \bigcup_{i=1}^{M} N^{\gamma}(C_{pq_i})} IC(C) \tag{13}$$

Then the set of images containing exclusively the query regions and neighbors is written as:

$$S_{result} = S_Q \setminus S_{XQ} \tag{14}$$

This set S_{result} constitutes the third and last level of relevant images. We note that $S_{XQ} \supset S_{NQ}$.

5.1.6 The three underlying query compositions

We proposed three compositions of multiple query in the previous subsections which provide different sets of relevant images S_{result}:

- positive query with $S_{result} = S_Q$
- negative query with $S_{result} = S_Q \setminus S_{NQ}$
- exclusive query with $S_{result} = S_Q \setminus S_{XQ}$

At each level the query formulation is more and restrictive and it is easy to see that they actually provide a decreasing set of relevant images (from relevant to most relevant), i.e.:

$$S_Q \supset S_Q \setminus S_{NQ} \supset S_Q \setminus S_{XQ} \tag{15}$$

From a same user query formulation, these three levels of search will be shown in the interface since they provide different information relevant to the user.

5.2 Implementation of retrieval by composition

We have to determine the different expressions of S_{result} (10, 12 and 14). The brute force approach would consist in testing each image in the database and see if it contains regions belonging to the PQCs (and their neighbors) but contain no region in any of the NQCs (and their neighbors). This would imply a number of comparisons close to 9995*5*(3*5 + 1*5)=1,000,000 comparisons for classic query of 3 PQCs and 1 NQC each having about 5 neighbor categories (a comparison being a simple integer value comparison to see if a region belongs to a given category).

We'll see that by using the properties of intersections and by preparing the database we can reduce dramatically this number of tests in a simple way.

The brute force approach (exhaustive search among all images) would consist in initialising S_{result} as the entire image database S_{DB}. It is adopted in current image retrieval systems.

Since S_{result} is expressed as intersections and substractions of image sets, the idea is to initialise S_{result} with one of the image sets and then discard images which don't belong to the other image sets. This initialisation avoids checking indivually each image of the database but rather start off from the very beginning with a set of potentially relevant images.

We first want to determine with S_Q, the set of images containing regions in the PQCs as defined in expression (8). At each search level, S_{result} will be reduced in the following order:

1. Positive query search: S_{result} is initialised as the set $\bigcup_{N^\gamma(C_{pq1})} IC(C)$. Then we discard images in S_{result} which do not belong to any of the other union categories ($i = 2, ..., M$) to obtain the intersections of S_Q as defined in expression (8). At this point, we have $S_{result} = S_Q$ (as in 10).

2. Negative query search: then to perform the substraction of S_{NQ} from S_{result}, we simply discard in S_{result} images which belong to any of the negative-query union categories ($i = 1, ..., R$) as written in 11. We get $S_{result} = S_Q \setminus S_{NQ}$ (as in 12).

3. Exclusive query search: to perform the substraction of S_{XQ} from S_{result}, we simply discard images in S_{result} which contain a region which is not in any of the positive-query nor their neighbors categories. We get $S_{result} = S_Q \setminus S_{NQ} \setminus S_{XQ} = S_Q \setminus S_{XQ}$ (as in 14).

So gradually, S_{result} is reduced from $\bigcup_{N^\gamma(C_{pq1})} IC(C)$ to $S_Q \setminus S_{XQ}$. By this approach, we'll see in section 6 that a significant fraction of the database is not accessed at all.

5.3 Index representation

In this section, we propose an indexing representation required by the retrieval strategy detailed just before. The retrieval strategy has shown that the initialisation of S_{result} allow to directly ignore a part of the irrelevant images from the database.

For a given category C_i, the initialisation requires to know its neighbor categories at any given range radius γ: it defines a first hash table $N(C_i)$.

$\forall i = 1, ..., P, N(C_i) = \{(N_1, d_1), (N_2, d_2), ..., (N_P, d_P)\}$ where the N_j are the neighbors of C_i

sorted by increasing prototype distance $d_j = d(C_i, N_j)$. By convention, N_1 is defined to be C_i itself (so $d_1 = 0$). For a given range γ at retrieval phase, the neighbors will be directly obtained as the subset $N^\gamma(C_i)$:
$N^\gamma(C_i) = \{(N_1, d_1), (N_2, d_2), ..., (N_K, d_K) \mid d_K \leq \gamma\}$

$N^\gamma(C)$ was first defined when region categories were generated.

For initialisation, we also need to know the list of images having a region in category C_i: it defines another hash table $IC(C_i) = \{I_k\}$. To process the other PQCs ($i = 2, ..., M$) we discard images from this initial S_{result} set which don't contain a region in any of them, then discard those which have a region in any of the NQCs ($i = 1, ..., R$). So we need to know, for any given image I, the list of categories its regions belong to: it defines another hash table $CI(I) = \{C_i^j\}$ which was already introduced in 5.1.1.

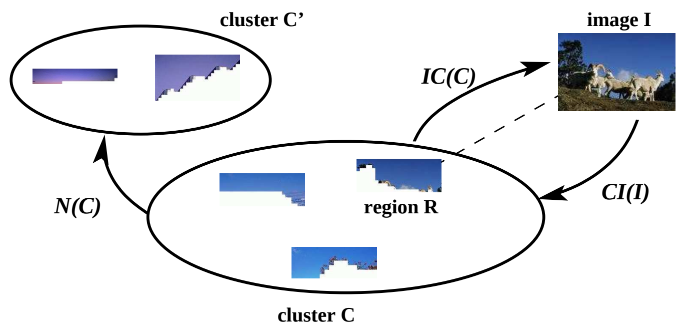

Figure 9: Hash tables N, CI, IC provide the following information: $N(C)$ is a neighbor category of C, $CI(I)$ is a category having a region of I and $IC(C)$ is an image containing a region (the region R) in category C.

The hash tables provide information between categories and images but do not involve regions. It is important to note that at retrieval time we don't deal with regions themselves but only with images and region categories, so that we don't have to individually access to the large number of regions in the database.

For a given range radius and a set of positive and negative query categories, the three query compositions detailed in the previous section are translated into accesses to hash tables N, CI, IC. We see here that search time is very fast since it only involves elementary operations on integers, unlike classic search approaches which require distance computations between multidimensional feature vectors.

6 Results and User interface

Results on the following parts of the system are presented: region extraction, generation of regions categories, relevance of retrieved images by composition and computational cost reduction of image retrieval scheme.

Note that no numerical evaluation of the retrieval performance of the system could be achieved due to the impossibility to build a meaningful groundtruth database for all possible scenarios.

The user interface will be presented to show how mutliple regions query can be simply formulated and will be illustrated by a query composition scenario.

Our system was tested with a 498 MHz Pentium PC running Linux. The test database consists of 9995 images from Corel Photostock. 50220 regions were extracted by the image segmentation scheme. 91 region categories were generated.

6.1 Region extraction

Even in complex natural scenes extracted regions present a coherent visual appearance and are generally intuitive for the user. The coarse segmentation proves its ability to integrate within regions areas formed with different shades of the same hue, strong textures, isolated spatial details. Such perceptual variability makes each region more specific against other regions in the database. Discarded regions (shown as small grey regions in examples of figure10) represent a very small percentage of image surfaces.

Examples of segmented images are presented in figure 10.

Segmenting a 384x256 image takes a around 3 seconds which is suitable for big image databases. 50,220 regions were automatically extracted from the 9995 images which yields an average of 5 regions per image.

6.2 Generation of region categories

Clustering the 50,220 regions mean colors takes 2 minutes and 30 seconds. 91 categories were automatically detected and their populations ranges from 112 regions to 2048 regions (average of 551 regions per category).

The most crowded categories correspond to low color saturation, i.e. black, white grey and a few other pale colors because they are located near the color cube center were data are the most dense. Conversely categories of high saturation (red, green, blue, yellow) are among the least crowded because they are near the color cube vertices and are statistically less frequent than low saturated colors. This makes saturated colors more discriminent for querying image.

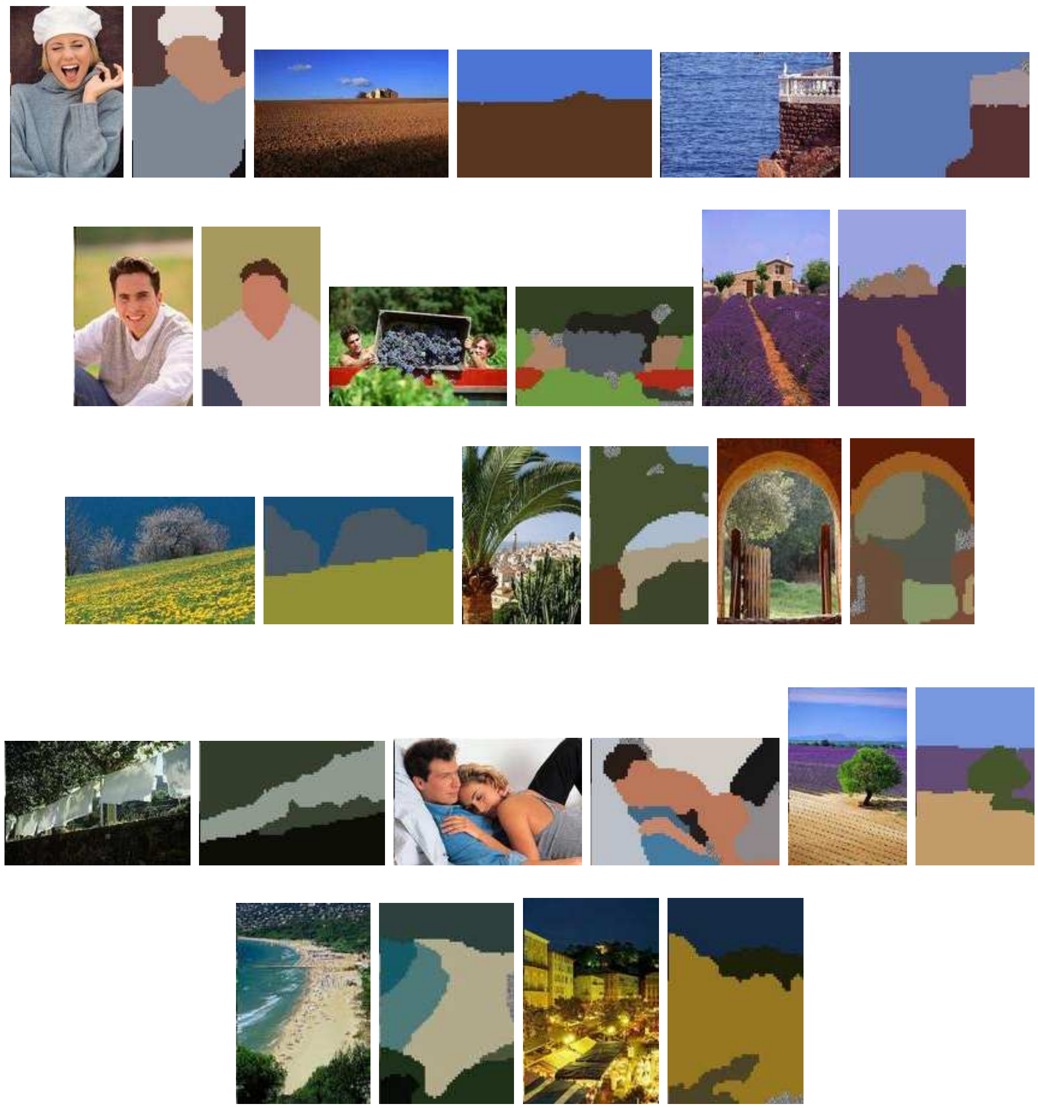

Figure 10: Illustration of coarse segmentation. Each original image is followed its segmented image of detected regions represented with mean color in RGB. Small discarded regions are shown in grey.

Figure 11 shows the content of 3 categories: orange, green and light blue. Since data are dense in the color space and the granularity is fine, the CA algorithm provides categories homogeneous in mean color as expected.

The intra-category variability is mainly due to the variety of different textures which can have a same mean color.

The 91 category representatives are shown in the query interface in figure 12. We remind that representatives just have a visualization purpose to identify each category in the interface. They are not involved in the search scheme.

The fine clustering granularity provides close categories which yield close representatives. Even when very close, two categories always differ in chrominance (it can be checked by looking at their overall content from the query interface).

6.3 User interface and retrieval relevance

The interface has two distinctive parts: one for query and one for results.

6.3.1 Query interface

Classic approach by QBE for image retrieval by global search (or region search) is tedious because it requires to browse many images before finding a satisfactory example image (or example region). For multiple region query this becomes even more tedious because we must repeat this process for each query region to formulate the full region query.

In our framework queries are expressed directly from the categories of similar regions. They are identified in the query interface by the thumbnail of their representatives which were defined in section 4.

They provide to the user an overview of the visual appearance of all regions he can expect to find (see figure 12).

In the query interface (fig. 12), all the 91 thumbnails of the regions representatives are displayed, below which 2 boxes can be checked to specify that this category constitutes a positive or negative query. It defines the query and negative-query lists $\{pq_1, pq_2, ..., pq_M\}$ and $\{nq_1, nq_2, ..., nq_R\}$ defined in section 5.1.1. At the bottom of the query window, a range box allows to adjust the range radius γ which will define interactively the neighbor categories of each query and negative-query category.

Each thumbnail of a representative can be clicked to display the content of the category it represents as illustrated before in figure 11. The thumbnails of all similar regions in the category are then shown in a new window. It helps the user know if the category corresponds precisely or not to the type of regions he wants to search.

6.3.2 Results interface

Given the selected range radius, each representative (of all positive query and negative query categories) is shown along with its neighbors defined within the range. The query formulation is expressed with logical operators (conjunction, disjunction and negations) of the PQCs

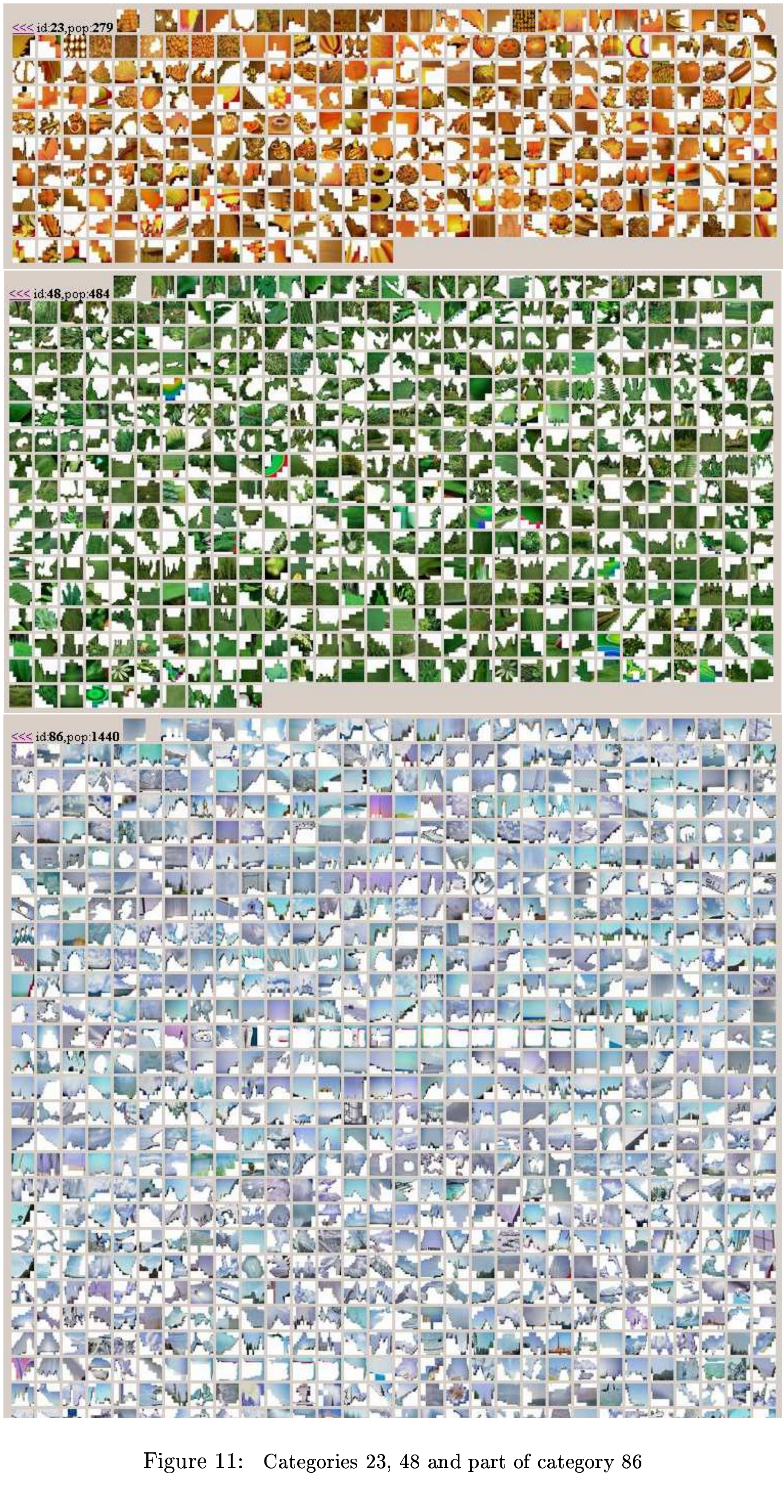

Figure 11: Categories 23, 48 and part of category 86

Figure 12: Query interface displays the 91 categories by their representatives: each category can be selected as a PQC or a NQC. The content of a category can be seen by clicking on a thumbnails. The range box allows the user to adjust the range radius for the definition of region categories.

and NQCs. OR between the neighbors (see equation 8). AND between query categories (see equation 9). $ANDNOT$ for negative query categories (see equation 11).

Figure 13 illustrates the query: "find images with sky-like and building-like regions but with no grass-like region". The user may want to find, for instance, cityscapes. A range value of 15 is selected for this query.

The system translates this composition into logical expression with the thumbnails of the PQCs (39 and 88) and the NQC (48) the query and negative query categories and their neighbors categories (see figure 14). Category 39 has 2 neighbors, category 88 has 3 and category 48 has 2.

The result images set is shown in three parts corresponding to the three query compositions from the most restrictive to the least: exclusive query, then negative query and positive query. A total of 191 images are retrieved for this query among the 9995.

To answer to the three query compositions defined in 5.1 we are interested in the sets of retrieved images $S_Q, S_Q \setminus S_{NQ}, S_Q \setminus S_{XQ}$. Since they are included in one another, we only present their differences as three disjoint subsets S_1, S_2, S_3. Figures 15, 16, 17 illustrate these sets for the query composition illustrated in figure 14.

The first set ($S_1 = S_Q \setminus S_{XQ}$) is the exclusive query set. Since exclusive query composition is very constraining, S_1 contains the few images which are very close to the query demand. It corresponds to a high retrieval precision, so it is shown in first position.
The second one ($S_2 = S_Q \setminus S_{NQ} \setminus S_1$) contains all images which have regions in the PQCs and none in the NQCs excluding images from S_1 already displayed. S_2 is the biggest set of retrieved images.
The last set ($S_3 = S_Q \cap S_{NQ}$) are the images which were rejected by the negative-query, i.e. images which have regions in the PQCs but also a region in the NQCs). It helps evaluate the relevance of the negative query to allow possibly refining the range radius.

In figure 15, S_1 contains 8 images among which 5 actually correspond to cityscapes with major regions of sky and grey buildings. Two of the other images are a beach scene and a nature landscape covered with grey ashes and a blue sky. And the last one does contain some green bush although a NQC was green. The reason is that the bush was segmented as a single highly textured region mixed with small white statues and a brown building. This false positive is due to a hard case of segmentation.

In images of S_2 (fig. 16), grey regions mostly correspond to buildings, monuments or rocks and blue regions to sky. Other grey regions are a ram, an ashes-covered tree or a collection car viewed as a complex region of black and white spots. Other blue regions are clothes, water or a blue wallpaper.

Figure 13: A query example: categories 39 and 88 are selected as positive query and 48 as negative. Range radius is set to 15.

Figure 14: Given the range 15, logical expression of query composition is automatically generated: neighbors are expressed as disjunctions, PQCs as conjunctions and NQCs as a negative conjunction.

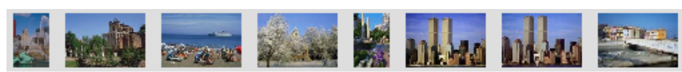

Figure 15: S_1, the first set of retrieved images corresponds to the exclusive query.

We see in figure 17 that images in S_3 do contain grass regions. They were correctly rejected by the *green* negative query category.

6.3.3 Query refinement

Modifying the query from the interface allows the user to refine the match of regions (by adjusting the range radius) and the match of images by composition (by modifying PQCs and NQCs).

The adjustment of the range value from one query session to another helps the user precising the types of regions of interest by decreasing or increasing the number of neighbors of both query and negative query categories. Indeed, since categories were defined as homogeneous, to identify regions of grass under different illuminations for instance, it may be interesting to expand the green detection by increasing the range value. On the other hand, to separate the grey of some buildings to the brownish-grey of rocks we shall decrease the range. The range is a parameter of refinement of retrieved types of regions.

The other way to refine the query is to iterate in the results by adding more negative query categories as we realise by seeing retrieved images that the absence of some categories may be discriminent for a particular search.

6.3.4 Relevance of retrieved images

As mentioned before, the evaluation of the precision of the retrieval by region categories composition can only be based on user satisfaction because the query scenarios are too

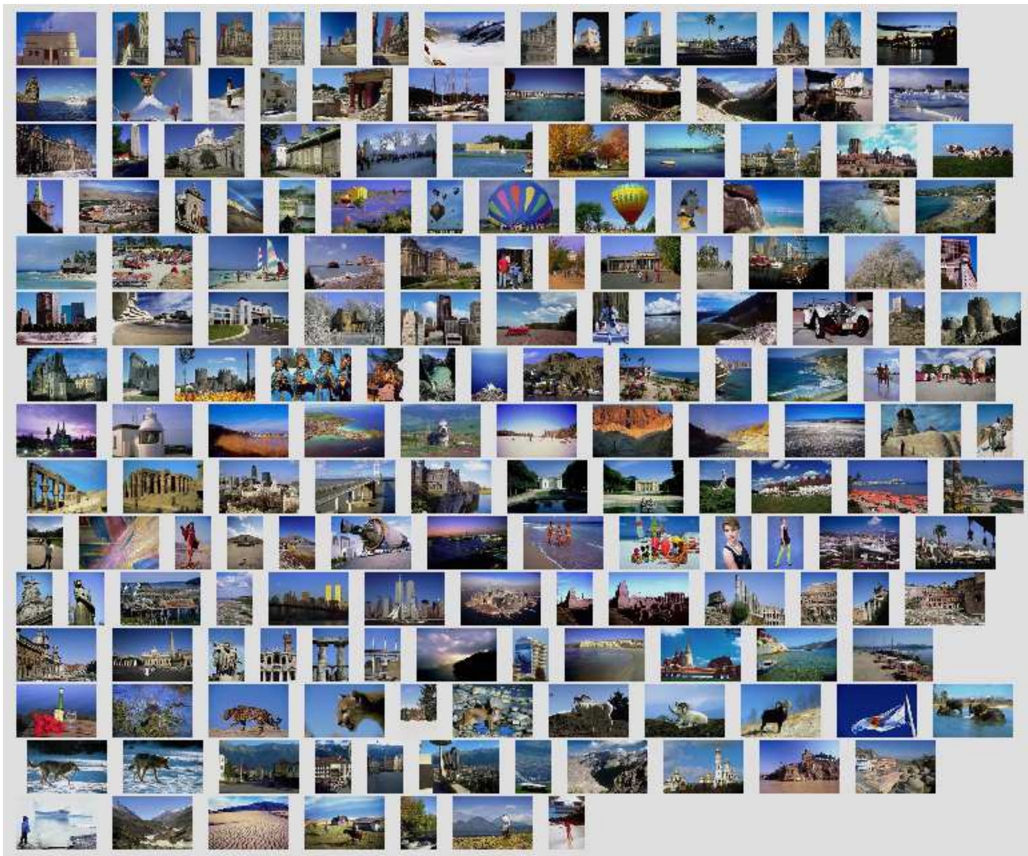

Figure 16: S_2, the second set of retrieved images corresponds to the negative query.

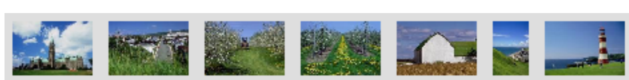

Figure 17: Set S_3 contains images rejected by negative query.

diverse in terms of range and composition and because it also depends on the user interaction in the query process especially if refinement is involved.

The relevance of matched regions relies on the region extraction and grouping schemes. In retrieved images, regions from the positive query categories are salient. For the user point of view, false positives among matched regions are few and correspond to hard segmentation cases (complex composite natural images). In this case, a detected region may not be meaningful even if its mean color corresponds to a query category.

Concerning the precision of composition matching in retrieved images, the simplicity of the indexing and retrieval scheme (comparison of category labels in image indexes) ensures a high user satisfaction. Indeed, in retrieved images, salient regions do satisfy the constraints of presence of regions from PQCs and the absence of regions of NQCs.

6.4 Image retrieval cost

6.4.1 Computational cost reduction

It should be noted that search time does not depend on descriptor dimension.
More generally in our framework, the choice of the visual descriptor and its associated distance only concerns the region grouping process which is performed "off-line". Indeed the only information used at search time are the precomputed distances between categories and the membership information about region categories and images. It only involves elementary operations on integer scalars.

Two other points in our search scheme reduce dramatically the number of accesses to the database entries at retrieval time. This scheme is:

- *not region exhaustive:* we don't directly deal with the $50,220$ regions anymore but only with their 91 category labels.
- *not image exhaustive:* only a fraction of the images in the database are accessed because the implementation of the hash tables provides direct access to potential relevant images

The access to a fraction of the image database is allowed by hash table IC when initialising the results set with $S_I = \bigcup_{N^\gamma(C_{pq_1})} IC(C)$, as we saw in 5.1. So the number of accessed image entries is the cardinality of S_I and it depends mainly on the range value. The higher the range value γ, the more numerous the number of neighbors of C_{pq_1} so the bigger the cardinality of S_I. On average on various query compositions (different range values, different types and numbers of query and negative categories), the fraction of accessed image entries $\mid S_I \mid / \mid S_{DB} \mid$ is around 12%.

6.4.2 Speed

Retrieval process takes up to 0.03 second for the most complex queries: high γ, many query and negative categories.

7 Concluding remarks

In this paper, we have presented a framework to retrieve images based on region categories composition. The system allows retrieving images by query composition like: "find images with regions of theses types and not like those types".

The originality of this approach relies on the grouping of similar regions into categories and has the following advantages:

- natural region range query by interactive definition of region categories
- query by image composition using regions categories
- efficient color indexing which results in very fast image retrieval

Although a very simple color region feature is used, the constraint of composition in retrieved images seems to express a certain underlying "visual semantic" in images.

We could investigate the integration in the framework of more specific individual regions features such as geometry (size and position) and photometry (such as ADCS [16] and also by considering color invariance [17]).

Contents

References

[1] M. Flickner and al., "Query by image and video content: the qbic system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[2] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *In SPIE Storage and Retrieval for Image and Video Databases*, vol. II, no. 2185, Feb. 1994.

[3] A. Gupta and al., "The virage image search engine: an open framework for image management,," *Storage and Retrieval for Image and Video Databases IV Proc. SPIE*, vol. 2670, 1996.

[4] A Smeulders, M. Worring, and al., "Content based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 12, pp. 1349–1380, 2000.

[5] C. Carson and al., "Blobworld: A system for region-based image indexing and retrieval," *Proc. of International Conference on Visual Information System (VIS'99)*, 1999.

[6] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia Systems*, vol. 7, no. 3, pp. 184–198, 1999.

[7] Jia Li James Z. Wang and Gio Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'01)*, 2001.

[8] S.F. Chang J.R. Smith, "Visualseek: A fully automated content-based image query system," in *ACM Multimedia Conference, Boston, MA, USA*, 1996.

[9] J. Fauqueur and Nozha Boujemaa, "Image retrieval by range query composition of region categories," *INRIA research report RR-4674*, 2002.

[10] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, 1997.

[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Functions*, Plenum, New York NY, 1981.

[12] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," in *Third International Conference on Visual Information Systems*. 1999, Springer.

[13] W.Y. Ma and B. Manjunath, "Edgeflow: A framework of boundary detection and image segmentation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 744–749, 1997.

[14] H. S. Sawhnay J. Hafner and al., "Efficient color histogram indexing for quadratic form distance functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 17, no. 7, pp. 729–736, July 1995.

[15] Theodosios Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin, 1977.

[16] J. Fauqueur and Nozha Boujemaa, "Region-based retrieval: Coarse segmentation with fine signature," *International Conference on Image Processing (ICIP'02)*, 2002.

[17] Th. Gevers and A. Smeulders, "Color based object recognition," *Pattern Recognition*, , no. 32, pp. 453–454, 1999.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399